

---

CHAPTER  
FOUR

---

CURRENT STATUS ON THE USE OF PARALLEL  
COMPUTING IN TURBULENT REACTING FLOW  
COMPUTATIONS INVOLVING SPRAYS,  
MONTE CARLO PDF & UNSTRUCTURED GRIDS

M. S. Raju

## 1 INTRODUCTION

The state of the art in multidimensional combustor modeling as evidenced by the level of sophistication employed in terms of modeling and numerical accuracy considerations, is also dictated by the available computer memory and turnaround times afforded by present-day computers. With the aim of advancing the current multi-dimensional computational tools used in the design of advanced technology combustors, a solution procedure is developed that combines the novelty of the coupled CFD/spray/scalar Monte Carlo PDF (Probability Density Function) computations on unstructured grids with the ability to run on parallel architectures. In this approach, the mean gas-phase velocity and turbulence fields are determined from a standard turbulence model, the joint composition of species and enthalpy from the solution of a modeled PDF transport equation, and a Lagrangian-based dilute spray model is used for the liquid-phase representation.

The gas-turbine combustor flows are often characterized by a complex interaction between various physical processes associated with the interaction between the liquid and gas phases, droplet vaporization, turbulent mixing, heat release associated with chemical kinetics, radiative heat transfer associated with highly absorbing and radiating species, among others [1]. The rate controlling processes often interact with each other at various disparate time

and length scales. In particular, turbulence plays an important role in determining the rates of mass and heat transfer, chemical reactions, and liquid phase evaporation in many practical combustion devices.

Most of the turbulence closure models for reactive flows have difficulty in treating nonlinear reaction rates [2-3]. The use of assumed shape PDF methods was found to provide reasonable predictions of pattern factors and  $\text{NO}_x$  emissions at the combustor exit [4]. However, their extension to multi-scalar chemistry becomes quite intractable. The solution procedure based on the modeled joint composition PDF transport equation has an advantage in that it treats the nonlinear reaction rates without any approximation. This approach holds the promise of modeling various important combustion phenomena relevant to practical combustion devices such as flame extinction and blow-off limits, and unburnt hydrocarbons (UHC), CO, and  $\text{NO}_x$  predictions [4].

With the aim of demonstrating the viability of the PDF approach to the modeling of practical combustion flows, we have undertaken the task of extending this technique to the modeling of sprays, unstructured grids, and parallel computing as a part of the NCC (National Combustion Code) development program [5-7]. NCC is being developed in the form of a collaborative effort between NASA LeRC, aircraft engine manufacturers, and several other government agencies [8].

The use of parallel computing offers enormous computational power and memory as it can make use of hundreds of processors in concert to solve a complex problem. The trend towards parallel computing is driven by two major developments: the widespread use of distributed computing and the recent advancements in MPPs (Massively Parallel Processor). The solver is designed to be massively parallel and automatically scales with the number of available processors. Also, the ability to perform the computations on unstructured meshes allows representation of complex geometries with relative ease. The grid generation time associated with gridding up practical combustor geometries, which tend to be very complex in shape and configuration, could be reduced considerably by making use of existing automated unstructured grid generators. The solver accommodates the use of an unstructured mesh with mixed elements: triangular and/or quadrilateral for 2D (two dimensional) geometries and tetrahedral for 3D. A solution procedure based on an unstructured grid formulation with parallel computing, is becoming an accepted practice for the numerical solution of complex multidimensional reacting flows (e.g., gas-turbine combustor flows) [8-10].

A complete overview of the overall solution method with a particular emphasis on the PDF and spray algorithms, parallelization, and several other numerical issues related to the coupling between the CFD, spray, and PDF solvers, is presented in this chapter. Some of the underlying differences between distributed computing and MPPs are discussed along with the underlying approaches to parallel programming involving Cray MPP Fortran and message passing libraries such as PVM (Parallel Virtual Machine) and MPI

(Message Passing Interface). The parallel performance of the three PDF, spray, and CFD modules is discussed for the case of a swirl-stabilized spray flame in both distributed and MPP computing environments. For a detailed presentation of the results and discussion, other than those discussed on the parallel performance, involving the application of this method to several flows involving swirl-stabilized spray flames and gaseous supersonic diffusion flames, the interested reader is referred to the published papers [1,7,24]. The chapter is concluded with some remarks on our ongoing research work and some suggestions for future research.

## 2 GOVERNING EQUATIONS FOR THE GAS PHASE

Here, we summarize the conservation equations for the gas phase in Eulerian coordinates derived for the multicontinua approach [11]. This is done for the purpose of identifying the interphase source terms arising from the exchanges of mass, momentum, and energy with the liquid phase.

The conservation of the mass leads to:

$$[\bar{\rho}V_c]_{,t} + [\bar{\rho}V_c u_i]_{,x_i} = s_{mlc} = \sum_k n_k m_k \quad (1)$$

For the conservation of the species, we have:

$$[\bar{\rho}V_c y_j]_{,t} + [\bar{\rho}V_c u_i y_j]_{,x_i} - [\bar{\rho}V_c D y_{j,x_i}]_{,x_i} - \bar{\rho}V_c \dot{w}_j = s_{mls} = \sum_k \epsilon_j n_k m_k \quad (2)$$

where

$$\sum_j \dot{w}_j = 0 \text{ and } \sum_j \epsilon_j = 1$$

For the momentum conservation, we have:

$$[\bar{\rho}V_c u_i]_{,t} + [\bar{\rho}V_c u_i u_j]_{,x_j} + [pV_c]_{,x_i} - [\theta V_c \tau_{ij}]_{,x_j} - [(1 - \theta)V_c \tau_{lij}]_{,x_j} = s_{mlm} = \sum_k n_k m_k u_{ki} - \sum_k \frac{4\pi}{3} \rho_k r_k^3 n_k u_{ki,t} \quad (3)$$

where  $\theta$  = the void fraction of the gas which is ratio of the equivalent volume of gas to a given volume of a gas and liquid mixture. For dilute sprays, the void fraction is assumed to be equal to one. The shear stress  $\tau_{ij}$  in Eq. (3) is given by:

$$\tau_{ij} = \mu[u_{i,x_j} + u_{j,x_i}] - \frac{2}{3}\delta_{ij}u_{i,x_j}$$

For the energy conservation, we have:

$$[\bar{\rho} V_c h]_{,t} + [\bar{\rho} V_c u_i h]_{,x_i} - [\theta V_c \lambda T_{,x_i}]_{,x_i} - [(1 - \theta) V_c \lambda_l T_{,x_i}]_{,x_i} - [\theta V_c p]_{,t} = s_{mle} = \sum_k n_k m_k (h_s - l_{k,eff}) \quad (4)$$

### 3 SCALAR JOINT PDF EQUATION

The transport equation for the density-weighted joint PDF of the compositions,  $\tilde{p}$ , is:

$$\begin{aligned} & [\bar{\rho} \tilde{p}]_{,t} + [\bar{\rho} \tilde{u}_i \tilde{p}]_{,x_i} + [\bar{\rho} w_\alpha(\underline{\psi}) \tilde{p}]_{,\psi_\alpha} = \\ & \{Transient\} \{Mean convection\} \{Chemical reactions\} \\ & - [\bar{\rho} < u_i'' | \underline{\psi} > \tilde{p}]_{,x_i} - [\bar{\rho} < \frac{1}{\rho} J_{i,x_i}^\alpha | \underline{\psi} > \tilde{p}]_{,\psi_\alpha} \\ & \{Turbulent convection\} \{Molecular mixing\} \\ & - [\bar{\rho} < \frac{1}{\rho} s_\alpha | \underline{\psi} > \tilde{p}]_{,\psi_\alpha} \\ & \{Liquid - phase contribution\} \end{aligned} \quad (5)$$

where

$w_\alpha$  = chemical source term for the  $\alpha$ -th composition variable,

$< u_i'' | \underline{\psi} >$  = conditional average of Favre velocity fluctuations,

$< \frac{1}{\rho} J_{i,x_i}^\alpha | \underline{\psi} >$  = conditional average of scalar dissipation, and

$< \frac{1}{\rho} s_\alpha | \underline{\psi} >$  = conditional average of spray source terms.

The terms on the left hand side of the above equation could be evaluated without any approximation but the terms on the right hand side of the equation require modeling. The first term on the right represents transport in physical space due to turbulent convection [3]. Since the joint PDF,  $\tilde{p}$ , contains no information on velocity, the conditional expectation of  $< u_i'' | \underline{\psi} >$  needs to be modeled. It is modeled based on a gradient-diffusion model with information supplied on the turbulent flow field from the flow solver [3].

$$- < u_i'' | \underline{\psi} > \tilde{p} = \Gamma_\phi \tilde{p}_{,x_i} \quad (6)$$

The fact that the turbulent convection is modeled as a gradient-diffusion makes the turbulent model no better than the  $k - \epsilon$  model. The uncertainties associated the use of a standard  $k - \epsilon$  turbulence model to swirling flows are well known [12]. Some of the modeling uncertainties associated with the use

of the standard  $k - \epsilon$  model would be addressed in our future studies with the implementation of a non-linear  $k - \epsilon$  developed for the modeling of swirling flows [12].

The second term on the right hand side represents transport in the scalar space due to molecular mixing. A mathematical description of the mixing process is rather complicated and the interested reader is referred to Ref. [3]. Molecular mixing is accounted for by making use of the relaxation to the ensemble mean submodel [2].

$$\langle \frac{1}{\rho} J_{i,x_i}^\alpha | \underline{\psi} \rangle = -C_\phi \omega (\phi_\alpha - \bar{\phi}_\alpha) \quad (7)$$

where  $\omega = \epsilon/k$ , and  $C_\phi$  is a constant. For a conserved scalar in a homogeneous turbulence, this model preserves the PDF shape during its decay but there is no relaxation to a Gaussian distribution [3]. However, the results of Ref. [4] indicate that the choice between the different widely-used mixing models is not critical in the distributed reaction regime of premixed combustion as long as the turbulent mixing frequencies are above 1000 Hz. Most of the practical combustors seem to operate in flame mixing frequencies of 1000 Hz and above. The application of this mixing model seemed to provide some satisfactory results when applied to flows representative of those encountered in the gas-turbine combustion [4].

The third term on the right hand side represents the contribution from the spray source terms. The conditional average is modeled based on the average values of species and enthalpy:

$$\langle \frac{1}{\rho} s_\alpha | \underline{\psi} \rangle = \frac{1}{\bar{\rho} \Delta V} \sum n_k m_k (\epsilon_{\alpha s} - \phi_\alpha) \quad (8)$$

where  $\phi_\alpha = y_\alpha$ ,  $\alpha = 1, 2, \dots, s = \sigma - 1$

$$\langle \frac{1}{\rho} s_\sigma | \underline{\psi} \rangle = \frac{1}{\bar{\rho} \Delta V} \sum n_k m_k (-l_{k,eff} + h_{k\sigma} - \phi_\sigma) \quad (9)$$

where  $\phi_\sigma = h$  and is defined by:

$$h = \sum_{i=1}^{\sigma-1} y_i h_i \quad (10)$$

where

$$h_i = h_{fi}^\circ + \int_{T_{ref}}^T C_{pi} y_i dT,$$

$$C_{pi} = \frac{R_u}{W_i} (A_{1i} + A_{2i}T + A_{3i}T^2 + A_{4i}T^3 + A_{5i}T^4),$$

$h_{fi}^\circ$  is the heat of formation of  $i$ th species,  $R_u$  is the universal gas constant,  $\epsilon_{\alpha s}$  is a mass fraction of the evaporating species at the droplet surface, and  $l_{k,eff}$

is the effective latent heat of vaporization as modified by the heat loss to the droplet interior:

$$l_{k,eff} = l_k + 4\pi \frac{\lambda_l r_k^2}{m_k} \left( \frac{\partial T_k}{\partial r} \right)_s \quad (11)$$

Here we assumed that the spray source terms could be evaluated independent of the fluctuations in the gas phase compositions of species and enthalpy. Eqs. (8)-(10) represent the modeled representation for the conditional averages of the spray contribution to the PDF transport equation.

#### 4 LIQUID PHASE EQUATIONS

The spray model is based on the multicontinua approach, which allows for resolution on a scale greater than the average spacing between two neighboring droplets [11]. A Lagrangian scheme is used for the liquid phase equations as it eliminates errors associated with numerical diffusion. The vaporization model of a polydisperse spray takes into account the transient effects associated with the droplet internal heating and the forced convection effects associated with droplet internal circulation and the phenomena associated with boundary layers and wakes formed in the intermediate droplet Reynolds number range [13]. The present formulation is based on a deterministic particle tracking method and on a dilute spray approximation which is applicable for flows where the droplet loading is low. Not considered in the present formulation are the effects associated with the droplet breakup, the droplet/shock interaction, the multi-component nature of liquid spray and the phenomena associated with dense spray effects and super-critical conditions. The spray method provided some favorable results when applied to both unsteady and steady state calculations [1,13-15].

For the particle position of the  $k$ th drop group, we have:

$$\frac{dx_{ik}}{dt} = u_{ik} \quad (12)$$

For the droplet velocity:

$$\frac{du_{ik}}{dt} = \frac{3}{16} \frac{C_D \mu_{gs} Re_k}{\rho_k r_k^2} [u_{ig} - u_{ik}] \quad (13)$$

where

$$Re_k = 2 \frac{r_k \rho_g}{\mu_{gs}} [(u_g - u_k) \cdot (u_g - u_k)]^{1/2} \quad (14)$$

$$C_D = \frac{24}{Re_k} \left( 1 + \frac{Re_k^{2/3}}{6} \right) \quad (15)$$

For droplet size, the droplet regression rate is determined from three different correlations depending upon the droplet-Reynolds-number range. When

$Re_k > 20$ , the regression rate is determined based on a gas-phase boundary-layer analysis [16] valid for Reynolds numbers in the intermediate range. The other two correlations valid when  $Re_k \leq 20$  are taken from Clift et al [17].

$$\begin{aligned} \frac{ds_k}{dt} &= -2 \frac{\mu_l}{\rho_k} \left[ \frac{2}{\pi} Re_k \right]^{1/2} f(B_k) \quad \text{if } Re_k > 20 \\ \frac{ds_k}{dt} &= -\frac{\mu_l}{\rho_k} \left[ 1 + (1 + Re_k)^{1/3} \right] Re_k^{0.077} \ln(1 + B_k) \\ &\quad \text{if } 1 < Re_k \leq 20 \end{aligned} \quad (16)$$

$$\frac{ds_k}{dt} = -\frac{\mu_l}{\rho_k} \left[ 1 + (1 + Re_k)^{1/3} \right] \ln(1 + B_k) \quad \text{if } Re_k < 1$$

where  $B_k$  is the Spalding transfer number defined in Eq. (22). The function  $f(B_k)$  is obtained from the solution of Emmon's problem. The range of validity of this function was extended in Raju and Sirignano [13] to consider the effects of droplet condensation.

The internal droplet temperature is determined based on a vortex model [16]. The governing equation for the internal droplet temperature is given by:

$$\frac{\partial T_k}{\partial t} = 17 \frac{\lambda_l}{C_{pl} \rho_l r_k^2} \left[ \alpha \frac{\partial^2 T_k}{\partial \alpha^2} + (1 + C(t) \alpha) \frac{\partial T_k}{\partial \alpha} \right] \quad (17)$$

where

$$C(t) = \frac{3}{17} \left[ \frac{C_{pl} \rho_l}{\lambda_l} \right] r_k \frac{dr_k}{dt} \quad (18)$$

where  $\alpha$  represents the coordinate normal to the streamsurface of a Hill's Vortex in the circulating fluid and  $C(t)$  represents a nondimensional form of the droplet regression rate. The initial and boundary conditions for Eq. (17) are given by

$$t = t_{injection}, \quad T_k = T_{k,o} \quad (19)$$

$$\alpha = 0, \quad \frac{\partial T_k}{\partial \alpha} = \frac{1}{17} \left[ \frac{C_{pl} \rho_l}{\lambda_l} \right] r_k^2 \frac{\partial T_k}{\partial t} \quad (20)$$

$$\alpha = 1, \quad \frac{\partial T_k}{\partial \alpha} = -\frac{3}{32} \frac{\rho_k}{\lambda_l} \left[ \frac{C_p (T_g - T_{ks})}{B_k} - l_k \right] \frac{ds_k}{dt} \quad (21)$$

where  $\alpha = 0$  refers to the vortex center and  $\alpha = 1$  refers to the droplet surface.

The Spalding transfer number is given by

$$B_k = \frac{C_p (T_g - T_{ks})}{l_{k,eff}} = \frac{(y_{fs} - y_f)}{(1 - y_{fs})} \quad (22)$$

$$y_{fs}^{-1} = 1 + \frac{M_a}{M_f} (\chi_{fs}^{-1} - 1) \quad (23)$$

where  $M_a$  is the molecular weight of the gas excluding fuel vapor.

Based on the assumption that phase equilibrium exists at the droplet surface, the Clausius-Clapeyron relationship yields

$$\chi_{fs} = \frac{P_n}{P} \exp \left[ \frac{l_k}{R_u} \left( \frac{1}{T_b} - \frac{1}{T_{ks}} \right) \right] \quad (24)$$

In Eq. (14) the molecular viscosity is evaluated at a reference temperature using Sutherland's equation

$$\mu(T_{ref}) = 1.4637 \cdot 10^{-6} \frac{T_{ref}^{3/2}}{T_{ref} + 120} \quad (25)$$

where

$$T_{ref} = \frac{1}{3}T_g + \frac{2}{3}T_{ks} \quad (26)$$

The droplets may evaporate, move along the wall surfaces, and/or reflect with reduced momentum upon droplet impingement with the combustor walls. In our present computations, subsequent to the droplet impingement with the walls, the droplets are assumed to flow along the wall surfaces with a velocity equal to that of the surrounding gas.

## 5 DETAILS OF DROPLET FUEL INJECTION

The success of any spray model depends a great deal on the specification of the appropriate injector exit conditions. However, a discussion involving the physics of liquid atomization is beyond the scope of this subject matter. In our present computations, the liquid fuel injection is simulated by introducing a discretized parcel of liquid mass in the form of spherical droplets at the beginning of every fuel-injection time step.

For certain cases, the fuel-injection time step,  $\Delta t_{ij}$ , needs to be determined based on the resolution permitted by the length and time scales associated with several governing parameters such as average grid spacing and average droplet spacing and velocity. However, our experience showed that for the case of a steady state solution, a time step based on the average droplet lifetime yields better convergence [13-15]. Its value typically ranges between 1 to 2 milli-seconds for the case of reacting flows.

The spray computations facilitate fuel injection through the use of a single fuel injector comprising of different holes [14-15]. However, multiple fuel injection in a steady state calculation could be simulated by simply assigning different initial conditions for the spatial locations of the droplet groups associated with each one of the different holes. For a polydisperse spray, the spray computations expect inputs for the number of droplet groups in a given stream and for the initial droplet locations and velocities. However, the number of droplets in a given group and their sizes could be either input directly or computed from a properly chosen function for the droplet size distribution.



The specified initial inputs should be representative of the integrated averages of the experimental conditions [1,7,14-15].

One correlation typical of those used for the droplet size distribution is taken from Ref. [18]:

$$\frac{dn}{n} = 4.21 \cdot 10^6 \left[ \frac{d}{d_{32}} \right]^{3.5} e^{-16.98 \left( \frac{d}{d_{32}} \right)^{0.4}} \frac{dd}{d_{32}} \quad (27)$$

where  $n$  is the total number of droplets and  $dn$  is the number of droplets in the size range between  $d$  and  $d + dd$ . The Sauter mean diameter,  $d_{32}$ , could be either specified or estimated from the following correlation [19]:

$$d_{32} = B_d \frac{2\pi\sigma_l}{\rho_g V_T^2} \lambda_m^* \quad (28)$$

where  $B_d$  is a constant,  $V_T$  is the average relative velocity between the liquid interface and the ambient gas, and  $\lambda_m^*$  is a function of the Taylor number,  $(\rho_l \sigma_l^2)/(\rho_g \mu_l^2 V_T^2)$ .

A typical droplet size distribution obtained from the above correlation in terms of the cumulative percentage of droplet number and mass as a function of the droplet diameter is shown in Fig. 1 [1].

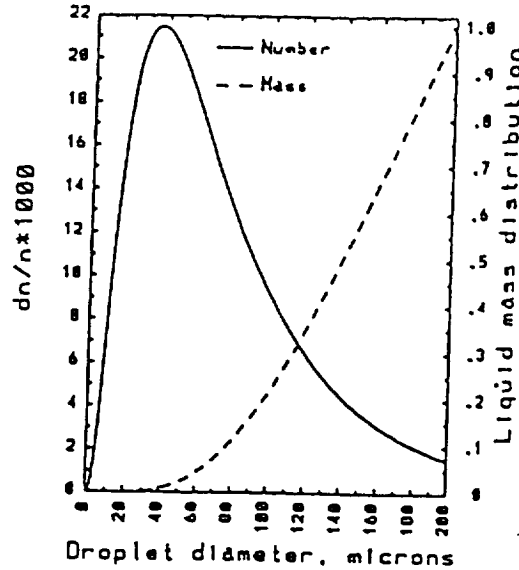


Figure 1 Droplet-size distribution.

## 6 CFD SOLUTION ALGORITHM

The gas phase mass and momentum conservation equations together with the standard  $k - \epsilon$  turbulence equations with wall functions are solved by making of a modified version of the Pratt and Whitney's CORSAIR - an unstructured CFD solver. It is a finite volume solver with an explicit fourth-stage Runge-Kutta scheme. Further details of the code can be found in Refs. [9-10].

## 7 PDF SOLUTION ALGORITHM

In order to facilitate the integration of the Monte Carlo PDF method in a finite-volume context, the volume integrals of convection and diffusion in Eq. (5) were first recast into surface integrals by means of a Gauss's theorem [20]. Partial integration of the PDF transport equation would yield:

$$\begin{aligned} \tilde{p}_p(\underline{\psi}, t + \Delta t) = & (1 - \frac{c_p \Delta t}{\bar{\rho} \Delta V}) \tilde{p}_p(\underline{\psi}, t) + \sum_n \frac{c_n \Delta t}{\bar{\rho} \Delta V} \tilde{p}_n(\underline{\psi}, t) \\ & - \Delta t [w_\alpha(\underline{\psi}) \tilde{p}]_{,\psi_\alpha} - \Delta t [\frac{1}{\rho} J_{i,x_i}^\alpha | \underline{\psi} > \tilde{p}]_{,\psi_\alpha} - \Delta t [\frac{1}{\rho} s_\alpha | \underline{\psi} > \tilde{p}]_{,\psi_\alpha} \end{aligned} \quad (29)$$

with subscript  $n$  refers to the  $n$ th-face of the computational cell. The coefficient  $c_n$  represents the transport by convection and diffusion through the  $n$ th face of the computational cell,  $p$ . The convection/diffusion coefficients in the above equation are determined by one of the following two expressions:

$$\begin{aligned} c_n &= \Gamma_\phi(\frac{2\underline{a}_n \cdot \underline{a}_n}{\Delta V_p + \Delta V_s}) + \max[0, -\bar{\rho} \underline{a}_n \cdot \underline{u}_n] \\ c_n &= \max[|0.5 \bar{\rho} \underline{a}_n \cdot \underline{u}_n|, \Gamma_\phi(\frac{2\underline{a}_n \cdot \underline{a}_n}{\Delta V_p + \Delta V_s})] - 0.5 \bar{\rho} \underline{a}_n \cdot \underline{u}_n \end{aligned}$$

and

$$c_p = \sum_n c_n$$

In both the above expressions for  $c_n$ , a cell-centered finite-volume derivative is used to describe the viscous fluxes; but an upwind differencing scheme is used for the convective fluxes in the first expression and a hybrid differencing scheme in the second.

### 7.1 Numerical Method Based on Approximate Factorization

The transport equation is solved by making use of an approximate factorization scheme [3]. Eq. (29) can be recast as

$$\begin{aligned} \tilde{p}_p(\underline{\psi}, t + \Delta t) = \\ (I + \Delta t R)((I + \Delta t S)(I + \Delta t M)(I + \Delta t T) \tilde{p}_p(\underline{\psi}, t) + O(\Delta t^2)) \end{aligned} \quad (30)$$

where  $I$  represents the unity operator and  $T$ ,  $M$ ,  $S$ , and  $R$  denote the operators associated with spatial transport, molecular mixing, spray, and chemical reactions, respectively. The operator is further split into a sequence of intermediate steps:

$$\tilde{p}_p^*(\underline{\psi}, t) = (I + \Delta t T) \tilde{p}_p(\underline{\psi}, t) \quad (31)$$

$$\tilde{p}_p^{**}(\underline{\psi}, t) = (I + \Delta t M) \tilde{p}_p^*(\underline{\psi}, t) \quad (32)$$

$$\tilde{p}_p^{***}(\underline{\psi}, t) = (I + \Delta t S) \tilde{p}_p^{**}(\underline{\psi}, t) \quad (33)$$

$$\tilde{p}_p(\underline{\psi}, t + \Delta t) = (I + \Delta t R) \tilde{p}_p^{***}(\underline{\psi}, t) \quad (34)$$

The operator splitting method provides the solution for the transport of  $\tilde{p}$  by making use of a Monte Carlo technique. In the Monte Carlo simulation the density weighted PDF at each grid cell is represented by an ensemble of  $N_m$  stochastic elements where the ensemble-averaged PDF over  $N_m$  delta functions replaces the average based on a continuous PDF [3].

$$\tilde{p}_{pm}(\psi) = \langle \tilde{p}_p(\psi) \rangle = \frac{1}{N_m} \sum_{n=1}^{N_m} \delta(\psi - \phi^n) \quad (35)$$

The discrete PDF  $\tilde{p}_{pm}(\psi)$  is defined in terms of  $N_m$  sample values of  $\phi^n$ ,  $n = 1, 2, 3 \dots N_m$ . The statistical error in this approximation is proportional to  $N_m^{-1/2}$ .

Using the operator splitting method, the solution for the PDF transport equation is obtained sequentially according to the intermediate steps given by Eqs. (31)-(34).

## 7.2 Convection/Diffusion Step

The first step associated with convection/diffusion is given by:

$$\begin{aligned} \tilde{p}_p^*(\underline{\psi}, t) &= (I + \Delta t T) \tilde{p}_p(\underline{\psi}, t) = \\ &= \left(1 - \frac{c_p \Delta t}{\bar{\rho} \Delta V}\right) \tilde{p}_p(\underline{\psi}, t) + \sum_n \frac{c_n \Delta t}{\bar{\rho} \Delta V} \tilde{p}_n(\underline{\psi}, t) \end{aligned} \quad (36)$$

This step is simulated by replacing a number of particles ( $=$  the nearest integer of  $\frac{c_p \Delta t N_m}{\bar{\rho} \Delta V}$ ) at  $\phi_p(t)$  by randomly selected particles at  $\phi_n(t)$ .

## 7.3 Numerical Issues Associated With Fixed Versus Variable Time Step

It is obvious from the above equation that a necessary criterion for stability requires satisfaction of  $\frac{c_p \Delta t}{\bar{\rho} \Delta V} < 1$ . When the computations are performed with a fixed time step, this criterion tends to be too restrictive for most applications: (1) Depending on the flow configuration, the allowable maximum time increment  $\Delta t$  is likely to be limited by a region of the flow field where convective fluxes dominate (such as close to injection holes). But in the main

stream, the flow is usually characterized by much lower velocities. (2) Resolution considerations require a higher concentration of the grid in certain regions of the flowfield than the others. For example, more grid lines are clustered in regions where boundary layers are formed. In such regions the allowable maximum time increment might be limited in a direction dominated by the largest of the diffusive fluxes as determined by  $\Gamma_\phi/\Delta x$ . This problem gets magnified if the cells also happen to be highly skewed.

Such restrictions on the allowable maximum time step could lead to a frozen condition when the Monte Carlo simulation is performed with a limited number of stochastic particles per cell. For clarity, let us consider the following criterion

$$N_m > \frac{\bar{\rho}\Delta V}{c_n\Delta t} \quad (37)$$

which has to be satisfied at all grid nodes. It is estimated that about  $10^3$  stochastic particles per cell are needed in order to avoid the so-called frozen condition for performing a typical 3-D gas-turbine combustor calculation. The frozen condition is referred to a state in which no transfer of stochastic particles takes place between the neighboring cells when  $N_m$  falls below a minimum required. Scheurlen et al [20] were the first ones to recognize the limitations associated with the use of a fixed time step in the Monte Carlo PDF computations.

However, our experience has shown that this problem can be overcome by introducing the concept of local time-stepping which is a convergence improvement technique widely used in many of the steady-state CFD computations. In this approach, the solution is advanced at a variable time step for different grid nodes. In our present computations, it is determined based on

$$\Delta t = \min(C_{tf}\Delta t_f, \frac{\rho\Delta V}{C_t(c_n + s_{mlc})}) \quad (38)$$

where  $C_{tf}$  and  $C_t$  are calibrated constants and were assigned the values of 4 and 2.5, respectively,  $\Delta t_f$  is the local time step obtained from the flow (CORSAIR) module, and  $s_{mlc} = \sum n_k m_k$ . The time step is chosen such that it permits transfer of enough particles across the boundaries of the neighboring cells while ensuring that the time step used in the PDF computations does not deviate very much from the time step used in the flow solver.

## 7.4 Molecular Mixing Step

The second step associated with molecular mixing is given by

$$\frac{d\phi_\alpha}{dt} = -C_\phi\omega(\phi_\alpha - \bar{\phi}_\alpha) \quad (39)$$

The solution for this equation is updated by:

$$\phi_{\alpha}^{**} = \phi_{\alpha}^* + (\phi_{\alpha}^* - \bar{\phi}_{\alpha}^*)e^{-C_{\phi}\omega\Delta t} \quad (40)$$

where  $C_{\phi}$  was assigned a value of 1.

### 7.5 Spray Step

The third step associated with the spray contribution is given by

$$\frac{d\phi_{\alpha}}{dt} = \frac{1}{\bar{\rho}\Delta V} \sum n_k m_k (\epsilon_{\alpha} - \phi_{\alpha}) \quad (41)$$

where  $\phi_{\alpha} = y_{\alpha}, \alpha = 1, 2, \dots, s = \sigma - 1$

$$\frac{d\phi_{\alpha}}{dt} = \frac{1}{\bar{\rho}\Delta V} \sum n_k m_k (-l_{k,eff} + h_{ks} - \phi_{\alpha}) \quad (42)$$

where  $\phi_{\sigma} = h$ .

The solution for the above equations is upgraded by a simple explicit scheme:

$$\phi_{\alpha}^{***} = \epsilon_{\alpha} \frac{\Delta t \sum n_k m_k}{\bar{\rho}\Delta V} + \phi_{\alpha}^{**} \left(1 - \frac{\Delta t \sum n_k m_k}{\bar{\rho}\Delta V}\right) \quad (43)$$

where  $\alpha \leq \sigma - 1$

$$\phi_{\alpha}^{***} = \frac{\Delta t \sum n_k m_k}{\bar{\rho}\Delta V} (-l_{k,eff} + h_{ks}) + \phi_{\alpha}^{**} \left(1 - \frac{\Delta t \sum n_k m_k}{\bar{\rho}\Delta V}\right) \quad (44)$$

where  $\alpha = \sigma$ .

After a new value for enthalpy is updated, the temperature is determined iteratively from the solution of Eq. (10).

### 7.6 Reaction Step

Finally, the fourth step associated with chemical reactions is given by:

$$\frac{d\phi_{\alpha}}{dt} = -\nu_f \frac{W_f}{\rho} A \left(\frac{\rho\phi_f}{W_f}\right)^a \left(\frac{\rho\phi_o}{W_o}\right)^b e^{-\left(\frac{E_a}{T}\right)} \quad (45)$$

where  $\phi_{\alpha} = y_f$ .

$$\frac{d\phi_{\alpha}}{dt} = -\nu_o \frac{W_o}{\rho} A \left(\frac{\rho\phi_f}{W_f}\right)^a \left(\frac{\rho\phi_o}{W_o}\right)^b e^{-\left(\frac{E_a}{T}\right)} \quad (46)$$

where  $\phi_{\alpha} = y_o$ .

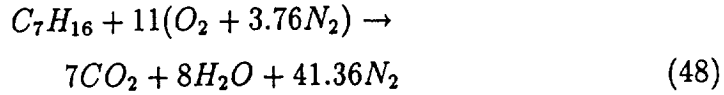
$$\frac{d\rho\phi_{\alpha}}{dt} = 0 \quad (47)$$

where  $\phi_{\alpha} = h$ .

The numerical solution for Eqs. (45)-(47) is integrated by an implicit Euler scheme [21]. The resulting non-linear algebraic equations are solved by the method of quasi-linearization [22].

## 7.7 Details of combustion chemistry

In this section, we present an example of how combustion chemistry is handled for the case of n-heptane when it is modeled by a single step global mechanism of Westbrook and Dryer [23]. The corresponding rate constants in Eqs. (45)-(46) are given by  $A = 0.286 \cdot 10^{+10}$ ,  $a = 0.25$ ,  $b = 1.25$ , and  $E_a = 0.151 \cdot 10^{+05}$ . This global combustion model is reported to provide adequate representation of temperature histories in flows not dominated by long ignition delay times. For example, the overall reaction representing the oxidation of the n-heptane fuel is given by



Because of the constant Schmidt number assumption made in the PDF formulation, based on atomic balance of the constituent species, the mass fractions of  $N_2$ ,  $CO_2$ , and  $H_2O$  can be shown to be related to the mass fractions of  $O_2$  and  $C_7H_{16}$  by the following expressions:

$$\begin{aligned} y_{H_2O} &= K_2 - K_1 K_2 y_{O_2} - K_2 y_{C_7H_{16}} \\ y_{CO_2} &= K_2 K_3 - K_1 K_2 K_3 y_{O_2} - K_2 K_3 y_{C_7H_{16}} \\ y_{N_2} &= 1 - K_2 - K_2 K_3 - y_{O_2}(1 - K_1 K_2 - K_1 K_2 K_3) - \\ &\quad y_{C_7H_{16}}(1 - K_2 - K_2 K_3) \end{aligned} \quad (49)$$

where  $K_1 = 4.29$ ,  $K_2 = 0.08943$ , and  $K_3 = 2.138$ .

Using Eq. (49) results in considerable savings in computational time as it reduces the number of variables in the PDF equation from five (four species and one energy) to three (two species and one energy).

## 7.8 Revolving Time-Weighted Averaging

It is noteworthy that although local time stepping seems to overcome some of the problems associated with the PDF computations, the application of the Monte Carlo method requires the use of a large number of particles because the statistical error associated with the Monte Carlo Method is proportional to the inverse square root of  $N_m$  which makes the use of the Monte Carlo method computationally very time consuming. However, a revolving averaging procedure used in our previous work [24] seems to alleviate the need for using a large number of stochastic particles,  $N_m$ , in any one given time step. In this averaging scheme, the solution provided to the CFD solver is based on an average of all the particles present over the last  $N_{av}$  time steps instead of an average solely based on the number of particles present in any one single time step. This approach seemed to provide smooth Monte Carlo solutions to the CFD solver and, thereby, improving the convergence of the coupled CFD and Monte Carlo computations. The reason for improvement

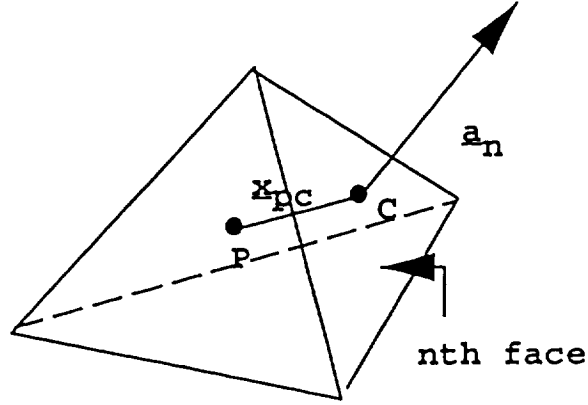


Figure 2 A vector illustration used in the particle search analysis.

could be attributed to an effective increase in the number of stochastic particles used in the computations from  $N_m$  to  $N_{av}N_m$ . Here, it is assumed that the solution contained within different iterations of the averaging procedure to be statistically independent of each other.

## 8 SPRAY SOLUTION ALGORITHM

In order to evaluate the initial conditions that are needed in the integration of the liquid phase equations, we first need to know the gas phase properties at each particle location. But in order to evaluate the gas phase properties, it is first necessary to identify the computational cell where a particle is located. It is a trivial task to search for the computational cell of the particle location in rectangular coordinates. However, a search for the particle location becomes a complicated problem when the computational cell is no longer rectangular in the physical domain. An efficient particle search algorithm is developed and implemented into the Lagrangian spray solver in order to facilitate particle movement in an unstructured grid of mixed elements. The search is initiated in the form of a local search from the computational cell of the previous time-step as the starting point. The location of the computational cell is determined by evaluating the dot product of  $\underline{x}_{pc} \cdot \underline{a}_n = |\underline{x}_{pc}| |\underline{a}_n| \cos(\phi)$ , where  $\underline{x}_{pc}$  is the vector defined by the particle location to the center of the n-face of the computational cell and  $\underline{a}_n$  is the outward area normal of the n-face as shown in Fig. 2, and  $\phi$  is the angle between the two vectors.

A simple test for the particle location requires that the dot product be negative over each and every one of the n-faces of the computational cell. If the test fails, the particle search is carried on over to the adjacent cells of

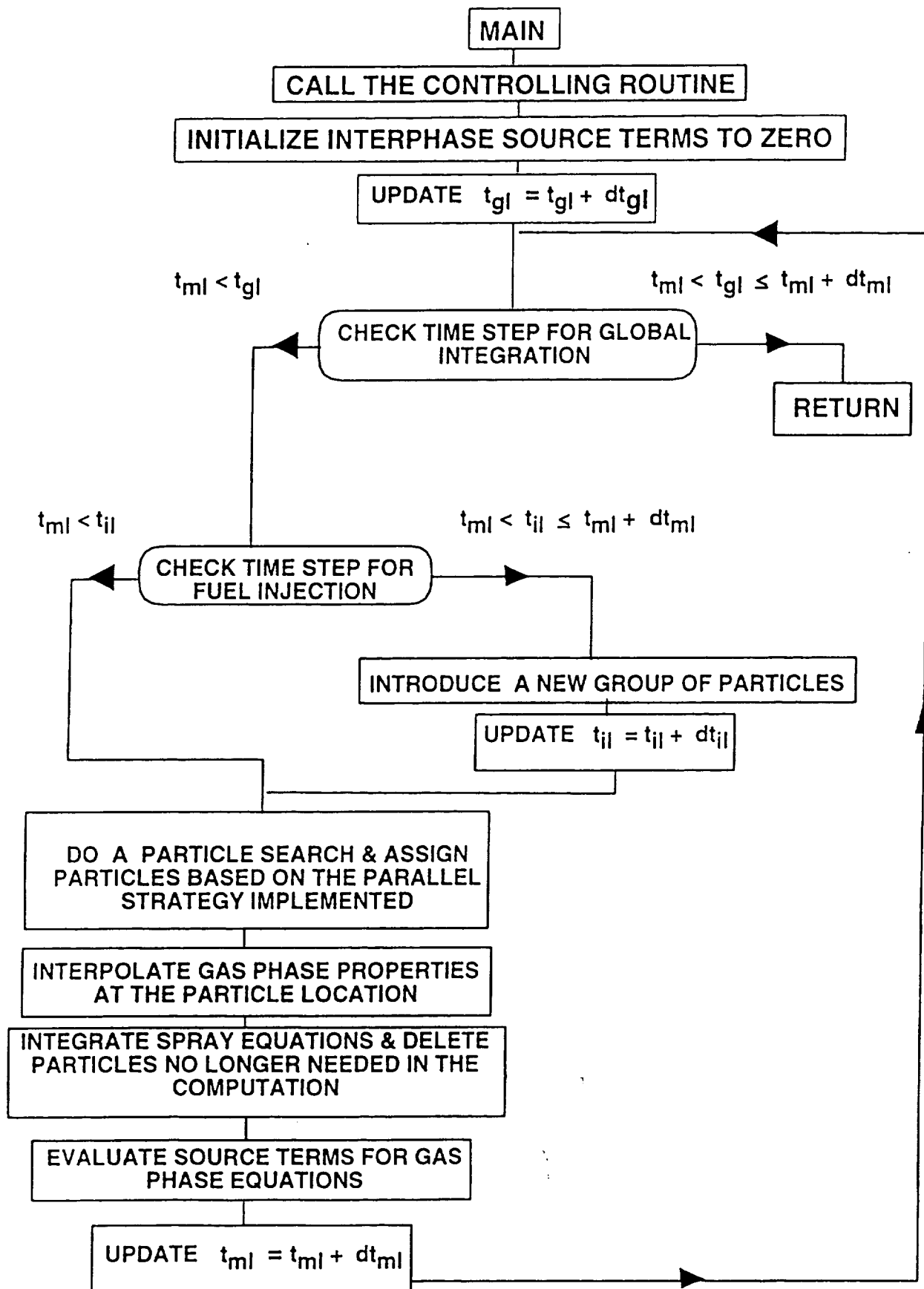


Figure 3 The flow structure of the spray code.



those faces over which the dot product turns out to be positive. Some of those n-faces might represent the boundaries of the computational domain while the others are the interfaces between two adjoining interior cells. The search is first carried on over to the adjacent interior cells in the direction pointed out by the positive sign of the dot products. The boundary conditions are implemented only after making sure that all the possibilities lead to a search outside of the computational boundaries. This implementation ensures against any inadvertent application of the boundary conditions before locating the correct interior cell.

After the gas phase properties at the particle location are known, the ordinary differential equations of particle position, size, and velocity are advanced by making use of a second-order accurate Runge-Kutta method. The partial differential equations governing the droplet internal temperature distribution are integrated by an Euler method. Finally, after the liquid phase equations are solved, the liquid phase source contributions to the gas phase equations are evaluated.

### 8.1 The Flow Structure of the Spray Code & Time-Averaging of the Interphase Source Terms of the Gas Phase Equations

The spray solver makes use of three different time steps -  $\Delta t_{ml}$  is the allowable time step,  $\Delta t_{gl}$  is the global time step, and  $\Delta t_{il}$  is the fuel injection time step.  $\Delta t_{ml}$  needs to be evaluated based on the smallest of the different time scales, which are associated with various rate controlling phenomena of a rapidly vaporizing droplet, such as those imposed by an average droplet lifetime, the local grid spacing and a relaxation time scale associated with droplet velocity among others. This restriction usually leads to a small time-step which typically has values in the neighborhood of 0.01 milli-seconds (ms). However, our experience has shown that the convergence for the steady state computations could be improved greatly by supplying the flow and PDF solvers with the interphase terms obtained from a time-averaging procedure, where the averaging is performed over an average lifetime of the droplets,  $\Delta t_{gl}$ . The variable,  $\Delta t_{gl}$ , has values in the neighborhood of 1 ms.

The averaging scheme could be explained better through the use of a flow chart shown in Fig. 3. The main spray solver is invoked by a call to the controlling routine which executes the following steps:

1. It first initializes the source terms to zero.
2. Checks to see if new particles need to be introduced.
3. Advances liquid phase equations over a pre-specified time step,  $\Delta t_{ml}$ , with calls to the following routines:
  - Does a particle search and assigns particles based on the parallel strategy implemented.

- Interpolates gas phase properties at the particle location.
  - Advances liquid phase equations and, also, deletes any particles that are no longer needed in the computations.
4. Evaluates the liquid phase source term contributions,  $S_{ml}$ , for use in the gas phase equations.
  5. Continues with steps (2) and (3) until the computations are completed over a global time step of  $\Delta t_{gl}$ .
  6. Returns control over to other solvers, e.g. flow or PDF, and supply them with source terms,  $S_{gl}$ , averaged over  $\Delta t_{gl}$ .

The time-averaged contribution of these source terms,  $S_{gl}$ , is given by:

$$S_{gl} = \sum_{m=1}^M \frac{\Delta t_{ml}}{\Delta t_{gl}} S_{ml} \quad (50)$$

where

$$\sum_{m=1}^M \Delta t_{ml} = \Delta t_{gl} \quad (51)$$

## 9 COUPLING BETWEEN THE THREE SOLVERS

For the PDF solver, the mean gas-phase velocity, turbulence diffusivity and frequency are provided as inputs from the CFD solver and the modeled spray source terms from the liquid-phase solver. And, in turn, the Monte-Carlo solver supplies the temperature and species fields to the other two solvers. The CFD code also receives the liquid-phase source terms as inputs from the spray solver. For the spray solver, the needed gas-phase velocity and scalar fields are supplied by the other two solvers. The liquid-phase, PDF, and CFD solvers are advanced sequentially in an iterative manner until a converged solution is obtained. It should also be noted that both the PDF and spray solvers are called once at every specified number of CFD iterations. All three PDF, CFD, SPRAY codes were coupled and parallelized in such a way in order to achieve maximum efficiency.

## 10 PARALLELIZATION

The trend towards the use of the parallel computing from that of serial vector machines is driven by several factors: the increased capabilities of RISC (Reduced Instruction Set Computing) processors, the limited increases in scalar/vector technology, the increased capabilities of network communication, the increased memory size with the availability of easily affordable DRAM (Dynamic Random Access Memory) chip storage capacity, and the

scalability of both memory size and problem requirements with the the number of processors. This lead to two major developments in parallel computing: the widespread use of distributed computing and MPPs.

The use of distributed computing is becoming widespread with the proliferation of workstation clusters which are tied into a network, and the availability of computer software such as PVM and MPI. For example, PVM was developed at ORNL (Oak Ridge National Laboratory). Both PVM and MPI provide a set of different Fortran and C++ library routines, which are available in the public domain and allow a network of heterogeneous computers to be used as a single large parallel computer. They provide the needed user-level message-passing interface for communicating between different PEs. Their main features include automatic data conversion, barrier synchronization, buffer management functions, task control functions, and data transmittal and receipt functions. Thus, large computational problems can be solved by using the aggregate power of many computers.

On the other hand, MPPs make use of hundreds of homogeneous processors in concert to solve a problem. For example, Cray T3D is a massively parallel computer with an aggregate of 64 PEs (Processor Elements). Each PE consists of a DEC Alpha chip 21064 with 8 Mwords of memory. The 64 PE Cray T3D delivers an aggregate peak performance of 19.2 Gflops on 64-bit data and supports a total of 512 Mwords of memory. The Cray T3D has 32 nodes configured in a three-dimensional (3-D) torus network topology with each node having two PEs. The topology permits fast interconnect network for communication and data movement. The unique features of MPPs also permit the support of special programming languages such as Cray MPP Fortran, which resembles in many ways to Fortran 90. Cray MPP Fortran provides several easier to implement programming tools such as shared memory functions, doshared directives, among several others. It is designed to support and exploit certain platform dependent hardware-specific intrinsics. Therefore, programs written in such programming languages provide significant performance gains over those written in Fortran 77 with PVM on platforms such as Cray T3D.

### 10.1 Gas phase domain decomposition methods

There are several ways to partition a grid. The most commonly used domain decomposition is referred to as 1-D partitioning, where the total domain is simply divided equally amongst the available PEs. Fig. 4 illustrates a simple example of the domain decomposition strategy adopted for the gas-phase computations. In this case, we assumed the number of available PEs to be equal to four. Any communication overhead associated with 1-D partitioning is limited to data transfer across the interfaces of the connecting subdomains.

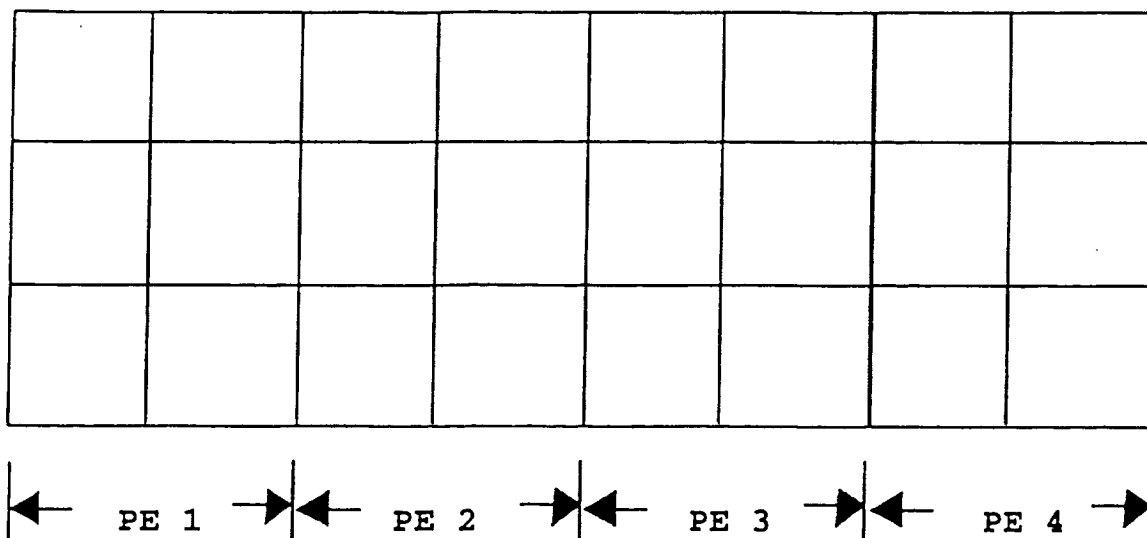


Figure 4 An illustration of the parallelization strategy employed in the gas flow computations.

## 10.2 Some of the programming differences between PVM & MPI versus Cray MPP Fortran

Here we highlight some of the basic differences and approaches to programming using PVM and Cray MPP Fortran. The concept of data sharing is to allow global variables, dummy arguments to be distributed across all PEs so that each PE can manipulate its share of data independently of other PEs. The challenge of programming is to keep data processing local to each PE and to keep PE to PE communication to a minimum while maintaining scalability. In distributed computing all data is private; no PE knows of any other PE's memory. But the need for communication between PEs couldn't be completely eliminated as for example during the numerical integration step, each PE need to know in general some information about what is contained in one or more layers of adjacent grid nodes located on the other side of the divided interface. This process is usually accomplished by preparing what are known as receive and send tables. Send tables contain information about what grid data need to be sent to the other PEs while receive tables contain information about what data need to be expected from the other PEs. During the information exchange process, the information contained in the send table is processed first by transmitting the required data to the other PEs before receiving the needed information from the other PEs. After receiving the data, it is stored in appropriate arrays. So for parallel implementation, a data preparation stage is clearly needed in terms of fetching, storing, and providing information on where to look for the stored data that was received.

PVM library also provides appropriate functions for use in the global information exchange and summation purposes. Such information might be needed in many places such as global mass preservation check, residuals evaluation, and propagating information on some reference variables. While PVM provides the needed user-level message-passing interface, it is up to the user to resolve several issues arising from the domain decomposition strategy adopted.

However, in Cray MPP Fortran, the need for using explicit communication calls is significantly minimized as it supports the use of shared memory. In shared memory, all PEs have access to the shared data without any need for invoking explicit communication calls regardless of where the memory is actually located. The use of shared data reduces the programming effort by a considerable degree as it eliminates the need for certain data preparation and information exchange stages associated with the use of private data. For a grid of mesh size  $I \times J$ , the shared data could be distributed such that each PE owns one block of contiguous elements,  $(I/N\$PES) \times J$ , where  $N\$PES$  is the total number of available PEs for a given application. In Fig. 4,  $N\$PES = 4$ ,  $I = 8$ , and  $J = 3$ . Similarly, arrays of different dimensions containing the  $i$ th dimension could be all blocked using 1-D partitioning. Instructions in a shared loop are divided up among processors, so each PE has subset of the entire instruction space. It should be kept in mind that only those arrays and variables that are needed to be accessed by other PEs, should be defined as the shared data while all others should be declared as private in order to achieve effective utilization of the available computer resources.

Even though Cray MPP Fortran provides a means for efficient implementation, its application is mainly limited to certain Cray computer platforms. Programming with PVM and MPI is gaining more ground as it offers a wider platform independence which is an important factor to consider in light of the fast pace at which work station cluster environment is changing.

### 10.3 Some basic guidelines to parallel implementation

There are several issues associated with the parallelization of the PDF and spray computations. The goal of the parallel implementation is to extract maximum parallelism so as to minimize the execution time for a given application on a specified number of processors [25]. Several types of overhead costs are associated with parallel implementation which include data dependency, communication, load imbalance, arithmetic, and memory overheads. Arithmetic overhead is referred to the extra arithmetic operations required by the parallel implementation and memory overhead refers to the extra memory needed. Excessive memory overhead reduces the size of a problem that can run on a given system and the other overheads result in performance degradation [25]. Any given application usually consists of several different phases that must be performed in certain sequential order. The degree of parallelism and data dependencies associated with each of the subtasks can vary widely [25]. The goal is to achieve maximum efficiency with a reasonable programming effort.

## 10.4 Parallel implementation of the PDF solver

Both the spray and kinetics steps of the PDF method lend themselves perfectly to parallel computing as no particle interaction of any kind occurs during their integration. During the mixing step the interaction between the particles is limited to only those particles present in a given cell. This step also lends itself to parallel computing without any associated overhead.

During the spatial transport step of the PDF solution method, particles are moved across the neighboring cells based on the computed values of the convection and turbulent-diffusion coefficients as determined from the numerical scheme used. The communication overhead in this step is limited to data transfer across the interfaces of the neighboring subdomains without any data dependency.

The combined overheads associated with parallel implementation is minimal since the time spent in the subroutines that deal with random number generation, convection and turbulent-diffusion, and boundary conditions, is only a small fraction of the total time that it takes for the entire PDF solution method. Therefore, a very high degree of parallelism could easily be achieved if enough care is exercised in distributing the spatial grid points uniformly amongst all the available PEs. Thus, the Monte Carlo simulation is ideally suited for parallel computing and the run time could be considerably minimized by performing the computations on a massively parallel computer.

## 10.5 Parallel implementation of the spray solver

In an approach, where an Eulerian scheme is employed for the gas phase computations and a Lagrangian scheme for the liquid phase computations, the spray computations are difficult to parallelize as the spray distribution tends to be both spatially non-uniform and temporally dynamic in nature for the reasons cited below:

- Most of the spray is usually confined to a small region near the atomizer location.
- The Lagrangian particles tend to move in and out of different parts of the computational domain processed by different PEs.
- Some new particles might be added to the computation at the time of fuel injection while some others might be taken out of computation. A particle is removed when it exits out of the computational boundaries or when it becomes small enough to the point where it is considered to be no longer needed in the computation.

Two different domain decomposition strategies were developed in order to explore their effectiveness on the parallel performance of the spray computations:

1. Strategy I: The Lagrangian particles are assigned uniformly amongst the available processors but the particle search and the computations involving the gas phase property evaluation at the particle location as well as the spray source terms, which are used in the CFD and PDF equations, were evaluated on the processor of the computational grid where the particle is located. This strategy leads to uniform load balancing during the integration of the liquid phase equations but may result in excessive message passing during the other operations.

This strategy yielded reasonable parallel performance when the computations were performed on a massively parallel computer like Cray T3D [1]. But its performance turned out to be rather poor when the computations were performed on the NASA LeRC LACE cluster [7]. The poor performance was identified to have resulted from the poor inter-processor communication capabilities of the workstation cluster. For that reason a second strategy was explored in order to improve the parallel performance of the spray code.

2. Strategy II: The Lagrangian particles are assigned to the processor of the computational grid where the particle is located. This strategy may lead to non-uniform load balancing during the integration of the liquid phase equations but is likely to result in less message passing since the inter-processor communications are limited to a single operation associated with the particle search.

## 10.6 Results and Discussion on the parallel performance

The applicability of the PDF approach for several test cases was documented in Refs. [1, 7, 24]. In a separate study, Chen et al [10] documented the performance of the CFD solver for several benchmark test cases. In this section, we only summarize the results of the parallel performance of the CFD, PDF, and spray solvers for two test cases.

The first case refers to the Cray MPP Fortran calculation of an open swirl-stabilized spray flame which was performed on a structured grid of  $60 \times 60 \times 3$  ( $=10,800$ ) nodes with a total of 2.7 million particles ( $=250$  particles/cell) requiring about 27 Mwords of computer memory for the major shared array allocation [1]. The computations were performed on the Cray T3D at NASA LeRC with the number of processors ranging between 8 to 32.

**Table 1. Cpu time (sec) per cycle versus number of PEs on Cray T3D.**

Solver	Characteristic	Number of processors		
		8	16	32
PDF solver	1 step/cycle	12.37	6.26	3.19
CFD solver	4 iterations/cycle	6.16	3.33	1.88
Spray solver (Strategy I)	50 steps/cycle	1.31	1.15	1.07

Table 1 summarizes the cpu times per cycle taken by the PDF, SPRAY, and CFD solvers. The cpu time for the PDF solver scales linearly with the number of processors, thereby, indicating the realization of a very high degree of parallelization. Even better scaling would be obtained if the i-th dimension of the grid is changed from 60 to 64, a power of 2. What is noteworthy is that the cpu times for the CFD solver also tend to scale linearly with the number of PEs used. The speed-up in the spray computations is considerably smaller than the other two solvers. Much of the slow-down in the spray computations is caused by the use of Cray MPP Fortran cdir\$ critical function, which was used to prevent any racing conditions from occurring during the summation of source-term contributions from different particles located in a given cell. By rewriting this part of the code the spray computations could be speeded up by a factor of 5 to 10.

The computations take about 2 hrs. of cpu time on 32 PEs to reach a converged solution. Based on our previous 2-D computations, it is expected for the cpu times on a CRAY T3D/32-PEs to be comparable with the performance of a CRAY Y-mP/1-PE.

The second case refers to the Fortran 77 with PVM calculation of a confined swirl-stabilized spray flame. The axisymmetric computations were performed on an unstructured grid of 3600 nodes and a total of 0.36 million Monte Carlo particles (=100 particles/cell). The computations were performed on NASA LeRC LACE cluster with the number of processors ranging between 2 to 16.

**Table 2. Cpu time (sec) per cycle versus number of PEs on LACE Cluster.**

Solver	Characteristic	Number of processors			
		2	4	8	16
PDF solver	1 step/cycle	2.08	1.16	0.67	0.42
CFD solver	5 steps/cycle	4.25	2.2	1.6	1.4
Spray solver (Strategy I)	100 steps/cycle	1.33	2.67	5.37	12.56
Spray solver (Strategy II)	100 steps/cycle	0.60	0.58	1.1	2.50

Table 2 summarizes the cpu times per cycle taken by the PDF, SPRAY, and CFD solvers versus the number of processors. The PDF solver shows good



parallel performance with an increase in the number of processors. The CFD solver also shows reasonable parallel performance but the gains seem to taper off more progressively with the increase in the number of PEs from 4 to 8. For the spray computations, the strategy II seems to result in a considerable improvement in parallel performance over the strategy I. Initially, with the strategy II, there is a slight performance gain with the increase in PEs from 2 to 4. However, there is a progressive deterioration in the performance when the number of PEs further increased from 4 to 16.

These results clearly demonstrate the need for a fast interconnect network for communication and data movement and the need for keeping the inter-processor communications to a minimum, especially when the computations are performed over a work station cluster environment. It is evident from the results that the maximum achievable parallel efficiency for the combined PDF/spray/CFD computations would be mostly constrained by the spray performance, especially when a Lagrangian representation is used for the spray and an Eulerian for the gas phase.

## 11 CONCLUDING REMARKS

A solution procedure has been outlined for the computation of turbulent spray flames on unstructured grids with parallel computing. The numerical method outlines several techniques designed to overcome some of the high computer time and storage limitations associated with the combined PDF/spray/CFD computations of practical combustor flows. The viability of the present method for its application to the modeling of practical combustion devices is also evident because of the ease with which grids could be generated for complex combustor geometries by making use of the commercially available interactive grid generation software like CFD-GEOM [26], which have the ability to generate interior grids from the data taken directly from the PATRAN-neutral files generated by several CAD/CAM packages.

There are several important aspects of spray combustion research that need to be addressed in order to provide better prediction tools needed in the design of advanced combustors. We conclude this chapter by making a few comments on our ongoing work and the planned research for the near future.

We are planning to extend our spray calculations to multi-component sprays under super-critical conditions. Since most of the aviation fuels are mostly multi-component, accurate representation of vaporization models becomes important in the prediction of some important combustion phenomena such as combustion instabilities, flame ignition, etc. [13]. Under super-critical conditions, flow field evolution is governed by both compressibility effects as well as variable inertial effects. Increase in pressure introduces both thermodynamic non-idealities and transport anomalies. Near the critical point, fluid properties exhibit liquid-like densities, gas-like diffusivities, and strongly pressure dependent solubilities. Surface tension and heat of vaporization approach zero and isothermal compressibility and constant pressure specific heat increase

significantly. These phenomena have a significant impact on the vaporization and overall dynamics associated with a given system. It is important to include the high pressure effects as most of the advanced technology combustors are planned to operate under elevated (near or above critical) pressure conditions.

There are several advantages to a Lagrangian representation for sprays:

- It is efficient on serial computers,
- It is the most widely used in dilute spray modeling, and
- Its solution is free of numerical diffusion.

As it is evident from our earlier discussion that it is very difficult to parallelize the overall calculation procedure, especially when it is used in conjunction with an Eulerian formulation for the gas phase and a Lagrangian formulation for the spray. The overall parallel performance of the combined solution could be vastly improved by developing a spray code based on an Eulerian formulation. The Eulerian approach offers several advantages over a Lagrangian formulation:

- It results in a highly efficient parallel algorithm for the combined PDF/spray/CFD solution,
- It is more convenient for including some of the dense spray effects, and
- It offers faster convergence to steady state solutions.

Not even the recent advances in parallel computing can provide the tremendous cpu time needed for a multi-species computation of a practical combustion flow. Therefore, it is important to develop reduced chemistry mechanisms that could be of interest in several important combustion phenomena such as emissions (NOX and CO), unburnt hydrocarbons (UHC), flame ignition and extinction limits. One approach that is developed as a part of the NCC effort is based on the ILDM (Intrinsic Low Dimensional Manifolds) method [10]. If the ILDM approach proved to be useful, significant cpu time savings could be realized by integrating the ILDM tables with the Monte Carlo PDF method.

## 12 ACKNOWLEDGEMENT

The research funding for this work was provided by NASA Lewis Research Center with Dr. N.-S. Liu acting as the technical monitor.

### 13 NOMENCLATURE

$A$	pre-exponent of an Arrhenius reaction rate term
$a$	non-unity exponent of an Arrhenius reaction rate term
$\underline{a}_n$	outward area normal vector of the $n$ th surface, $m^2$
$B_k$	Spalding transfer number
$b$	non-unity exponent of an Arrhenius reaction rate term
$C_D$	drag coefficient
$C_p$	specific heat, $J/(kg\ K)$
$C_\phi$	a constant in Eq. (39)
$c_n$	convection/diffusion coefficient of the $n$ th face, $kg/s$
$D$	turbulent diffusion coefficient, $m^2/s$
$d$	drop diameter, $m$
$E_a$	activation energy of an Arrhenius reaction rate term
$h$	specific enthalpy, $J/kg$
$J_i^\alpha$	diffusive mass flux vector, $kg/ms$
$k$	turbulence kinetic energy, $m^2/s^2$
$l_k$	latent heat of evaporation, $J/kg$
$l_{k,eff}$	effective latent heat of evaporation, $J/kg$ (defined in Eq. (11))
$M_i$	molecular weight of $i$ th species, $kg/kg\text{-mole}$
$m_k$	droplet vaporization rate, $kg/s$
$m_{ko}$	initial mass flow rate associated with $k$ th droplet group
$N_{av}$	number of time steps employed in the PDF time-averaging scheme
$N_f$	number of surfaces contained in a given computational cell
$N_m$	total number of Monte Carlo particles per grid cell
$N_p$	total number of computational cells
$n_k$	number of droplets in $k$ th group
$P$	pressure, $N/m^2$
$Pr$	Prandtl number
$p$	joint scalar PDF
$R_u$	gas constant, $J/(kg\ K)$
$Re$	Reynolds number
$r_k$	droplet radius, $m$
$r_{ko}$	initial drop radial location, $m$
$s_k$	droplet radius squared, $r_k^2$ , $m^2$
$s_{mlc}$	liquid source contribution of the gas-phase continuity equation
$s_{mle}$	liquid source contribution of the gas-phase energy equation
$s_{mlm}$	liquid source contribution of the gas-phase momentum equations
$s_{mls}$	liquid source contribution of the gas-phase species equations
$s_\alpha$	liquid source contribution of the $\alpha$ variable
$T$	temperature, $K$
$t$	time, $s$
$u_i$	$i$ th velocity component, $m/s$
$u_{ik}$	$i$ th velocity component of $k$ th drop group, $m/s$
$V_c$	volume of the computational cell, $m^3$

$w_\alpha$	chemical reaction rate, 1/s
$\dot{w}_j$	gas phase chemical reaction rate, 1/s
$\dot{w}_j$	gas phase chemical reaction rate, 1/s
$x_i$	Cartesian coordinate in the $i$ th direction, m
$y_j$	mass fraction of $j$ th species
$\underline{x}$	spatial vector
$\chi$	mole fraction
$\Delta t$	local time step used in the PDF computations, s
$\Delta t_f$	local time step in the flow solver, s
$\Delta t_{gl}$	global time step in the spray solver, s
$\Delta t_{il}$	fuel injection time step, s
$\Delta t_{ml}$	allowable time step in the spray solver, s
$\Delta V$	computational cell volume, $m^3$
$\delta$	Dirac-delta function
$\epsilon$	rate of turbulence dissipation, $m^2/s^3$
$\epsilon_j$	species mass fraction at the droplet surface
$\epsilon_{\alpha s}$	species mass fraction at the droplet surface
$\Gamma_\phi$	turbulent diffusion coefficient, kg/ms
$\lambda$	thermal conductivity, J/(ms K)
$\mu$	dynamic viscosity, kg/ms
$\omega$	turbulence frequency, 1/s
$\phi$	represents a set of scalars of the joint PDF
$\underline{\psi}$	independent composition space
$\rho$	density, $kg/m^3$
$\sigma$	dimensionality of $\underline{\psi}$ -space
$\tau$	stress tensor term
$\theta$	void fraction

## Subscripts

$f$	represents conditions associated with fuel
$g$	global or gas-phase
$i$	index for the coordinate or species components
$j$	index for the species component
$k$	droplet group or liquid phase
$l$	liquid phase
$m$	conditions associated with $N_m$
$n$	$n$ th-face of the computational cell
$o$	initial conditions or oxidizer
$p$	conditions associated with the properties of a grid cell
$s$	represents conditions at the droplet surface or adjacent computational cell
$t$	conditions associated with time
$\alpha$	index for the scalar component of the joint PDF equation
,	partial differentiation with respect to the variable followed by it

## Superscripts

- ~ Favre averaging
- time averaging or average based on the Monte Carlo particles present in a given cell
- // fluctuations

## 14 REFERENCES

1. M.S. Raju, Application of Scalar Monte Carlo Probability Density Function Method For Turbulent Spray Flames, Numerical Heat Transfer, Part A, vol. 30, pp. 753-777, 1996.
2. R. Borghi, Turbulent Combustion Modeling, Prog. Energy Combust. Sci., vol. 14, pp. 245-292, 1988.
3. S.B. Pope, PDF Methods for Turbulent Reactive Flows, Prg. Energy Combust. Sci., vol. 11, pp. 119-192, 1985.
4. S.M. Correa, Development and Assessment of Turbulence-Chemistry Models in Highly Strained Non-Premixed Flames, AFOSR/NA Contractor Report, 110 Duncan Avenue, Bolling AFB, DC 20332-0001, 31 October, 1994.
5. M.S. Raju, LSPRAY - a Lagrangian Spray Solver - User's Manual, NASA/CR-97-206240, NASA Lewis Research Center, Cleveland, Ohio, November, 1997.
6. M.S. Raju, EUPDF - an Eulerian-Based Monte Carlo Probability Density Function (PDF) Solver - User's Manual, NASA/CR-1998-20401, NASA Lewis Research Center, Cleveland, Ohio, April, 1998.
7. M.S. Raju, Combined Scalar Monte Carlo PDF/CFD Computations of Spray Flames on Unstructured Grids With Parallel Computing, AIAA/ASME/SAE/ASEE 33rd Joint Propulsion Conference, Seattle, Wash., July 6-9, 1997.
8. N.S. Liu and R.M. Stubbs, Preview of National Combustion Code, AIAA 97-3114, AIAA/ASME/SAE/ASEE 33rd Joint Propulsion Conference, Seattle, Wash., July 6-9, 1997.
9. R. Ryder, CORSAIR User's Manual: Version 1.0, SID: Y965, Pratt and Whitney Engineering, United Technologies Corporation, 25 January 1993.
10. K.-H. Chen, A.T. Norris, A. Quealy, and N.-S. Liu, Benchmark Test Cases For the National Combustion Code, AIAA 98-3855, AIAA/ASME/SAE/ASEE 34th Joint Propulsion Conference, Cleveland, Ohio, July 13-15, 1998.
11. W.A. Sirignano, Fluid Dynamics of Sprays, Journal of Fluids Engineering, vol. 115, no. 3, pp. 345-378, September 1993.

12. T.-H. Shih, K.-H. Chen, and N.-S. Liu, "A Non-Linear  $k - \epsilon$  Model for Turbulent shear flows," AIAA/ASME/SAE/ASEE 34th Joint Propulsion Conference, Cleveland, Ohio, July 13-15, 1998.
13. M.S. Raju and W.A. Sirignano, Multi-Component Spray Computations in a Modified Centerbody Combustor, *Journal of Propulsion and Power*, vol. 6, no. 2, pp. 97-105, 1990.
14. M.S. Raju, AGNI-3D: A Computer Code for the Three-Dimensional Modeling of a Wankel Engine, *Computers in Engine Technology: Proceedings IMechE*, London, United Kingdom, pp. 27-37, 1991.
15. M.S. Raju, Heat Transfer and Performance Characteristics of a Dual-Ignition Wankel Engine, *Journal of Engines Sect. 3*, SAE Trans., vol. 101, pp. 466-509, 1992.
16. Tong, A.Y. and Sirignano, W.A., "Multi-component Transient Droplet Vaporization With Internal Circulation: Integral Formulation and Approximate Solution," *Numerical Heat Transfer*, Vol. 10, pp. 253-278, 1986.
17. Clift, R., Grace, J.R., and Weber, M.E., *Bubbles, Drops, and Particles*, Academic, New York, 1978.
18. El Banhawy, Y., and Whitelaw, J.H., "Calculation of the Flow Properties of a Confined Kerosene-Spray Flame," *AIAA J.*, Vol. 18, No. 12, PP. 1503-1510, 1980.
19. Bracco, F.V., "Modelling of Engine Sprays," SAE paper 850394, 1985.
20. M. Scheurlen, B. Noll, and S. Wittig, Application of Monte Carlo Simulation For Three-Dimensional Flows, AGARD-CP-510: CFD Techniques For Propulsion Applications, February 1992.
21. D.A. Anderson, J.C. Tannehill, and R.H. Fletcher, *Computational Fluid Mechanics and Heat Transfer: Series in Computational Methods in Mechanics and Thermal Sciences*, Hemisphere Publishing Corporation, Washington D.C., U.S.A., 1984.
22. M.S. Raju, W.Q. Liu, and C.K. Law, A Formulation of Combined Forced and Free Convection Past Horizontal and Vertical Surfaces, *Int. J. Heat and Mass Transfer*, vol. 27, pp. 2215-2224, 1984.
23. C.K. Westbrook and F.L. Dryer, Chemical Kinetic Modelling of Hydrocarbon Combustion, *Progress in Energy and Combustion Science*, vol. 10, no. 1, pp. 1-57, 1984.
24. A.T. Hsu, Y.-L.P. Tsai, and M.S. Raju, A Probability Density Function Approach for Compressible Turbulent Reacting Flows, *AIAA Journal*, vol. 32, no. 7, pp. 1407-1415, 1994.
25. J.S. Ryan and S.K. Weeratunga, Parallel Computation of 3-D Navier-Stokes Flowfields for Supersonic Vehicles, AIAA 93-0064, AIAA 31st Aerospace Sciences Meeting, Reno, Nevada, 1993.

26. CFD-GEOM, Version 2.0, CFD Research Corporation, 3325 Triana Blvd., Huntsville, Alabama 35805, July 1996.

